USBORNE

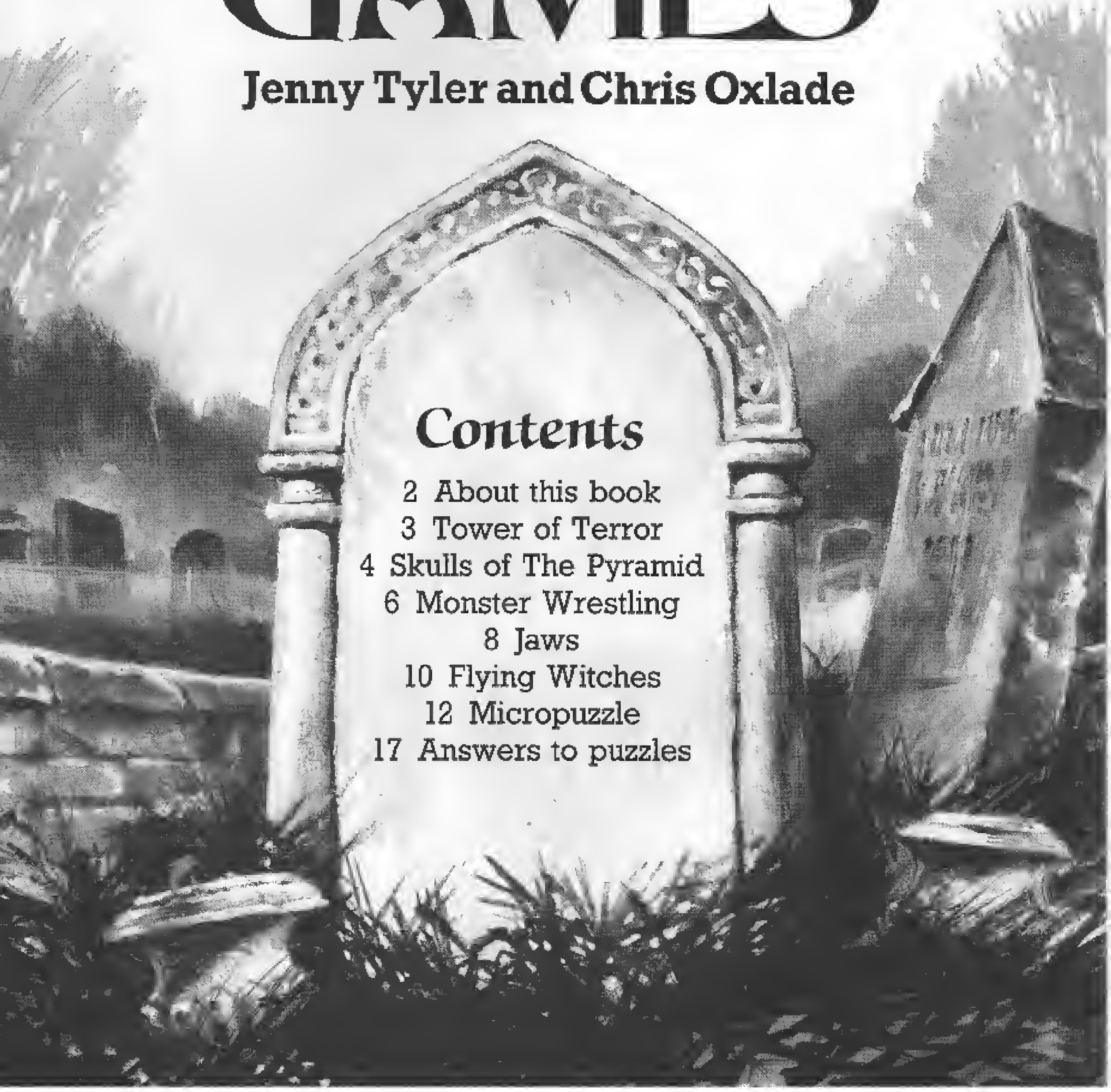
WEIRD COMPUTER GAMES

.....FOR...COMMODORE 64...VIC 2

....TRS 80 32K ...BBC...ELECTRON.

WEIRD COMPUTER GAMES

Jenny Tyler and Chris Oxlade



Contents

About this book

The programs in this book are written in a standard version of BASIC and there are conversion lines to type in for most of the main types of home computers. Look down the left-hand side of the program for the symbol for your computer and then look at the list of changes for the correct version of that line. The symbols for the various computers are as follows:

▲Commodore 64 and VIC 20
★BBC and Electron
✡Spectrum
●Apple
■TRS-80 (extended BASIC version)

About the games

The games in this book are very simple. They are intended to help you get used to your computer and to the BASIC language by typing in listings, debugging them and seeing how they work. The programs do not contain graphics or sound as these vary so much from computer to computer, but you can try adding these.

You can change and adapt the games as much as you like. There are suggestions for ways of doing this next to each program and you can experiment with your own ideas as well. This way you can use the games in this book as a basis for longer, more complicated games of your own.

Typing and running the programs

Remember, even short programs can be quite difficult and time-consuming to type in correctly. Check each line as you go. It is so easy to make mistakes, even if you are quite experienced. When you have typed in the whole listing, check it again, making sure you haven't missed any lines, spaces or punctuation.

To start the game, type RUN. Read the introduction to the game first so that you have some idea of what you are supposed to do before you start. If the program doesn't work properly, it is quite likely that there is a mistake in it somewhere, so LIST the program and check again.

When the game is over, the computer may ask if you want to play again or say something like BREAK in 200, in which case you must type RUN to play again.

Changing the speed

Some games depend on the speed of both your reactions and your computer. You may find you need to adjust the speed. You will find instructions for doing this next to the program listing.

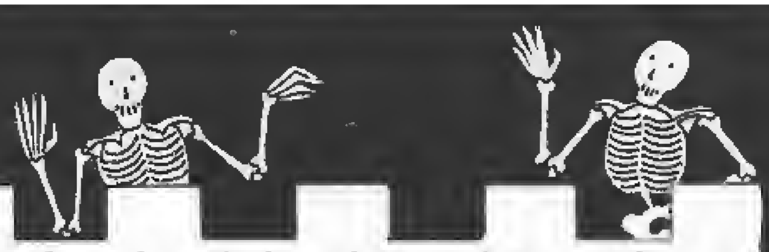
Micropuzzle was written by Les Howarth and Monster Wrestling by Adrian Hall.

Illustrated by Rob McCaig, Sue Walliker, Martin Newton and Graham Smith.

Designed by Roger Priddy

First published in 1984 by Usborne Publishing Ltd, 20 Garrick Street, London WC2E 9BJ.

Tower of Terror

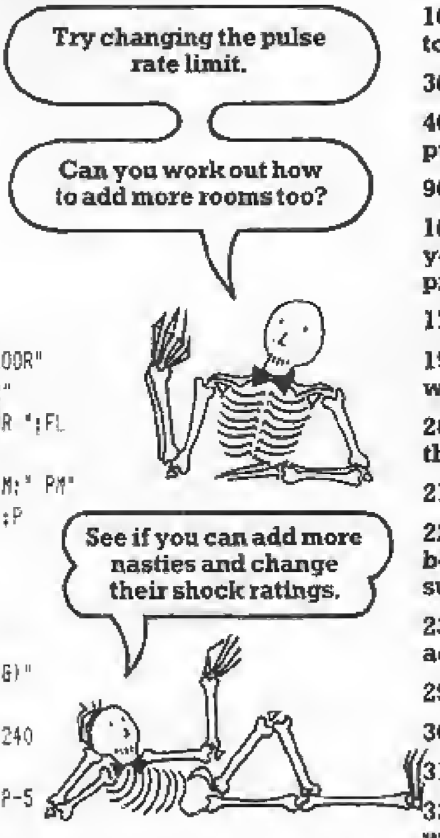


Your mouth is dry, your legs are shaking and your heart is thumping – you've entered the Tower of Terror . . .

Press G to move through the rooms. Agh!! there's a skeleton, then a ghost, then a headless axeman! With each fresh shock your pulse rate rockets.

Will you go on (G), or retreat (R) to recover a little? Watch the time – you've only got until midnight to reach the top of the Tower and the coveted Treasure. Watch your pulse rate too. Madness takes over when it reaches 150 and nothing can stop you leaping to your doom out of the window.

```
     10 GOSUB 470
     20 LET R$="GOOD LUCK"
     30 LET RM=0
■✡  40 LET H=9:LET M=INT(RND(1)*10)+10
     50 LET P=50
●▲  60 CLS:PRINT:PRINT
     70 PRINT "TOWER OF TERROR"
     80 PRINT "==============="
     90 PRINT:PRINT R$
    100 LET R$="":LET FL=INT(RM/5)
    110 LET R=RM-FL*5+1
    120 PRINT:PRINT "YOU ARE ON"
    130 IF FL=0 THEN PRINT "THE GROUND FLOOR"
    140 IF FL=6 THEN PRINT "THE TOP FLOOR"
    150 IF FL>0 AND FL<6 THEN PRINT "FLOOR ";FL
    160 PRINT "IN ROOM ";R
    170 PRINT:PRINT "THE TIME IS ";H;".";M;" PM"
    180 PRINT:PRINT "YOUR PULSE RATE IS ";P
    190 LET GF=0
    200 IF RM=30 THEN GOTO 350
■✡ 210 LET TR=INT(RND(1)*9+1)
■✡ 220 IF RND(1)>.6 THEN GOSUB 420
    230 PRINT:PRINT "RETREAT OR GO ON (R/G)"
★●▲ 240 LET I$=INKEY$
    250 IF I$<>"G" AND I$<>"R" THEN GOTO 240
    260 IF I$="G" THEN GOSUB 400
    270 IF I$="R" THEN LET RM=RM-1:LET P=P-5
    280 IF RM=-1 THEN LET RM=0
■✡ 290 LET M=M+INT(RND(1)*3+1):IF M>59 THEN LET M=M-60:LET H=H+1
    300 IF H=12 THEN GOTO 360
    310 IF P>150 THEN GOTO 380
    320 IF P<40 THEN LET P=40
■✡ 330 IF FL=TR AND RND(1)>0.5 THEN GOSUB 520
    340 GOTO 60
    350 PRINT "WELL DONE ":STOP
    360 PRINT:PRINT "IT'S MIDNIGHT!"
    370 PRINT:PRINT "TOO LATE!":STOP
    380 PRINT "YOU HAVE GONE MAD AND"
    390 PRINT "LEAPT FROM A WINDOW!":STOP
    400 IF GF=1 THEN LET P=P+S*2:LET R$="AAAHHHH!"
    410 LET P=P-1:LET RM=RM+1:RETURN
■✡ 420 LET TY=INT(RND(1)*3+1)
    430 LET W$=G$(TY)
■✡ 440 LET S=INT(RND(1)*5)+FL+TY*2
    450 PRINT:PRINT "AHEAD YOU SEE A ";W$
    460 LET GF=1:RETURN
 ✡ 470 DIM G$(3)
    480 LET G$(1)="SKELETON"
    490 LET G$(2)="GHOST"
    500 LET G$(3)="HEADLESS AXEMAN"
    510 RETURN
    520 LET R$="YOU FELL THROUGH A TRAP DOOR!":LET RM=RM-5:LET P=P+10:RETURN
```


Try changing the pulse rate limit.
Can you work out how to add more rooms too?
See if you can add more nasties and change their shock ratings.

What is the Treasure of Tower and how did it get there . . .? Perhaps you can add your ideas to the program.

How it works

10: Goes to initialization section to read in data.

30: RM is the room you're in.

40-50: Set the starting time and pulse rate.

90: Prints any comment stored in R$.

100-160: Work out floor and room you are in from value of RM and print this information.

170-180: Print other information.

190: GF is a flag which is set when there is a nasty ahead.

200: Checks if you've reached the last room.

210: Selects room for trap door.

220: Random chance of there being something ahead. Goes to subroutine if there is.

230-280: Get player's input and act on it.

290: Increases time.

300: Tests for midnight.

310: Tests if pulse rate is too high.

330: Checks if you are in room with trap door.

400-410: Move forward routine. Increase pulse if there's a nasty.

420-460: Select a type of nasty and its shock rating.

470-510: Read in data for start of game (initialization).

520: Trap door subroutine.

Conversion lines

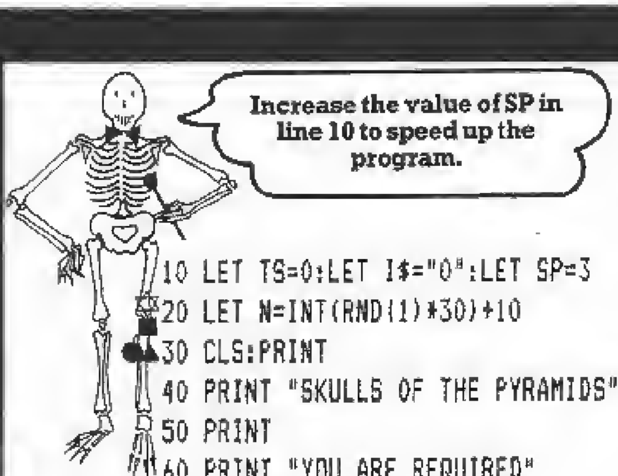
```
✡ 40,210,220,290,330,420,440 Replace
      RND(1) with RND
■ 40,210,220,290,330,420,440 Replace
      RND(1) with RND(0)
▲ 60 PRINT CHR$(147):PRINT:PRINT
● 60 HOME:PRINT:PRINT
★ 240 LET I$=INKEY$(0)
▲ 240 GET I$
● 240 I$="":IF PEEK(-16384)>127
       THEN GET I$
✡ 470 DIM G$(3,15)
```

Skulls of the Pyramid

A mind without a body – that's all you've been since that terrible accident all those years ago, during the building of The Pyramid.

It takes all your mental energy to satisfy the Evil Spirit Master's demands. Every day he forces a number into your mind (if only you could turn him off!). You must will skulls to fall off The Pyramid onto the numbered stones below in an attempt to score that number.

Press 1, 2, 3 or 4 to release a skull. You must use five skulls each go – no more, no less. Score the right number and rid yourself of the Curse of The Pyramid.



Increase the value of SP in line 10 to speed up the program.

```
    10 LET TS=0:LET I$="0":LET SP=3
    20 LET N=INT(RND(1)*30)+10
  ●▲30 CLS:PRINT
    40 PRINT "SKULLS OF THE PYRAMIDS"
    50 PRINT
    60 PRINT "YOU ARE REQUIRED"
    70 PRINT "TO GET ";N:PRINT
    80 LET K=3000:GOSUB 540
  ●▲90 CLS:LET Y=2
   100 FOR I=1 TO 4
   110 LET A$=STR$(I):LET X=6+2*I
   120 GOSUB 560
   130 NEXT I
   140 LET A$="^":FOR I=1 TO 5
   150 LET Y=2+2*I
   160 FOR J=1 TO I+3
   170 LET X=7-I+2*J
   180 GOSUB 560
   190 NEXT J
   200 NEXT I
   210 LET Y=14
   220 FOR I=1 TO 9:LET A$=STR$(I)
   230 LET X=1+2*I:GOSUB 560
   240 NEXT I
   250 LET S=0
   260 FOR I=1 TO 5
   270 LET A$="CHOOSE ROW"
   280 LET X=0:LET Y=0:GOSUB 560
★●▲290 LET I$=INKEY$
   300 IF I$=""THEN GOTO 290
   310 LET R=VAL(I$)
   320 IF R<1 OR R>4 THEN GOTO 290
   330 LET Y=3:LET K=400:LET X=6+2*R
   340 LET F=0
```

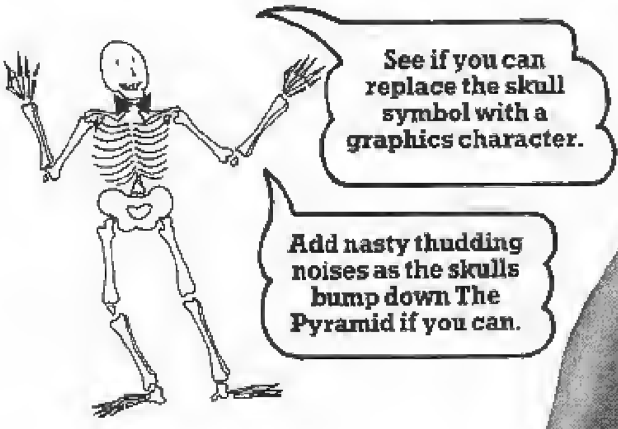
How the program works

10: Sets variables at beginning of game.

20: Chooses number that player must score.

30-80: Print starting message and wait for player to see it.

90-240: Print out pyramid and the numbers at the top and bottom of it.



See if you can replace the skull symbol with a graphics character.

Add nasty thudding noises as the skulls bump down The Pyramid if you can.

250: Sets score to zero.

260: Sets number of skulls.

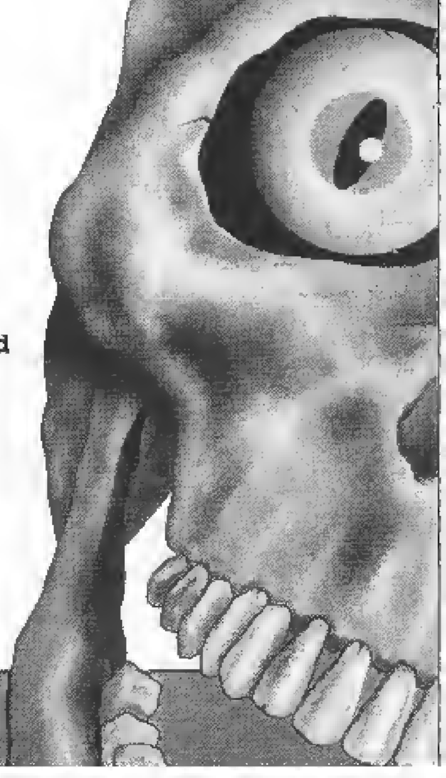
270-280: Print message top left of screen.

290-300: Wait for key to be pressed.

310-320: Check if key pressed is between 1 and 4 and go back to wait again if it isn't.

330: X and Y are position of skull. This line sets them at the top of the pyramid.

340: F is a flag which records whether previous move was a sideways bounce.



```
    350 LET A$="O"
    360 GOSUB 560
    370 GOSUB 540
    380 LET A$=" ":GOSUB 560
  ■✡390 IF F=0 AND Y<13 THEN LET X=X+SGN(RND(1)-.5)
    400 LET Y=Y+1:LET F=1-F
    410 IF Y<14 THEN GOTO 350
    420 LET S=S+(X-1)/2
    430 LET X=15:LET Y=0
    440 LET A$=STR$(I)+":"+STR$(S)
    450 GOSUB 560
    460 NEXT I
    470 LET K=1000:GOSUB 540
    480 LET D=ABS(S-N):LET TS=TS+D
  ●▲490 CLS:PRINT:PRINT
    500 PRINT "YOUR RATING IS NOW ";TS
    510 IF D>1 THEN LET K=3000:GOSUB 540:GOTO 20
    520 PRINT "AND YOU CAN GO FREE!"
    530 STOP
    540 FOR T=1 TO K STEP SP:NEXT T
    550 RETURN
✡■●▲560 PRINT TAB(X,Y);A$
    570 RETURN
```

350-370: Print the skull and wait.

380: Remove the skull.

390: Choose which way to bounce (if value of F allows sideways bounce).

400: Moves skull down one row.

410: Checks if skull has reached the bottom.

420: Works out score depending on column skull finishes in.

430-450: Print out score.

470: Waits.

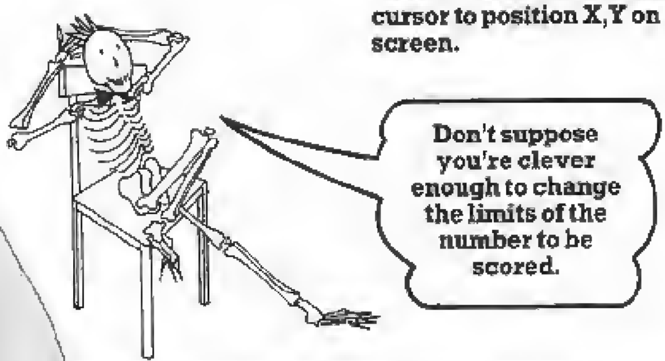
480-500: Add difference between score and number required to total score. Clear screen and print score.

510: If difference is greater than 1, goes back for another go.

520: Prints winning message.

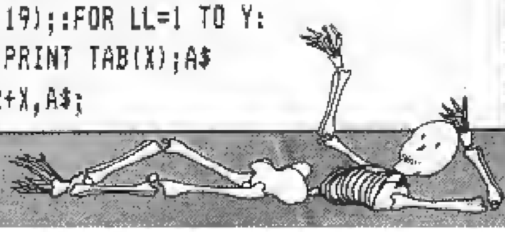
540-550: Subroutine to make computer pause for length of time depending on K.

560-570: Subroutine to move cursor to position X,Y on the screen.


Don't suppose you're clever enough to change the limits of the number to be scored.

Conversion lines

```
✡20,390 Replace RND(1) with RND
■20,390 Replace RND(1) with RND(0)
●30,90,490 Replace CLS with HOME
▲30,90,490 Replace CLS with PRINT CHR$(147)
★290 LET I$=INKEY$(0)
●290 I$="":IF PEEK(-16384)>127 THEN GET I$
▲290 GET I$
✡560 PRINT AT Y,X;A$
●560 VTAB(Y+1):HTAB(X+1):PRINT A$
▲560 PRINT CHR$(19);:FOR LL=1 TO Y:
     PRINT:NEXT:PRINT TAB(X);A$
■560 PRINT@ Y*32+X,A$;
```



Monster Wrestling

Monster wrestling is a sport for lunatics! Which doesn't say much for you – the brain in charge of this hulk of bone and muscle which is about to take on some of the nastiest monsters in the Universe.

As brain, you must do a lot of quick and accurate calculations. You must work out the muscular effort required to hold off the monster, for instance, and this involves multiplying the size of the monster by the distance it is away from you.

If the numbers look too difficult, you can press the Panic Button (key P). You must then work out how much adrenalin the body needs to survive the crisis, by dividing heartbeat increase required by oxygen supply. Take care though, over-use of the Panic Button puts too much strain on the heart and will eventually cause a black out.

To live to fight another match, you must survive 12 rounds against the monster.

```
   20 LET P=0
   30 LET K=3
●▲40 CLS
   50 LET X=1
   60 LET Y=6
   70 LET N=-1
   75 LET N=N+1
■☆80 LET G=INT(RND(1)*Y+X)
■☆90 LET I=INT(RND(1)*K+K)
  100 LET Y=Y+0.5
  110 LET X=X+0.5
  120 LET K=K+0.5
  130 PRINT
  140 PRINT
  150 PRINT "SIZE OF MONSTER: ";
  160 PRINT G
  170 PRINT
  180 PRINT "DISTANCE AWAY: ";
  190 PRINT I
  200 PRINT
  210 PRINT "MUSCULAR EFFORT? ";
  220 GOSUB 570
  230 IF Z<>G*I THEN GOTO 320
●▲240 CLS
  250 PRINT "MONSTER KEPT AT BAY"
  260 IF N<11 THEN GOTO 75
  270 PRINT "PHEW!!!!-THE MONSTER"
  280 PRINT "IS TIRED AND HAS GONE TO"
  290 PRINT "LOOK FOR ANOTHER VICTIM."
  300 PRINT "YOU SURVIVE TO TELL THE TALE!"
  310 STOP
●▲320 CLS
  330 PRINT "YOU HAVE BEEN CRUSHED"
  340 PRINT "TO A PULP IN THE"
  350 PRINT "MONSTER'S HUGE ARMS"
  360 PRINT
  370 PRINT "YOU SURVIVED ";N;" ROUNDS"
```

How the program works

20: Sets number of panics used to zero.

30-60: Set upper limits for size of monster and distance away.

70: Sets number of rounds.

80: Sets monster size.

90: Sets distance away.

100-120: Increase upper limits for size and distance by 0.5 each round.

210-220: Ask for answer and go to subroutine to deal with it.

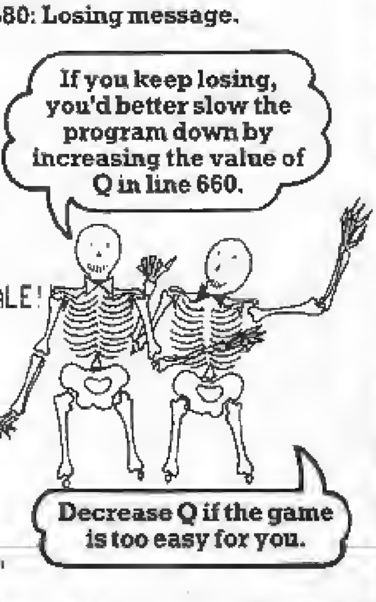
230: If answer incorrect, jumps down program for losing message.

250: Message if round won.

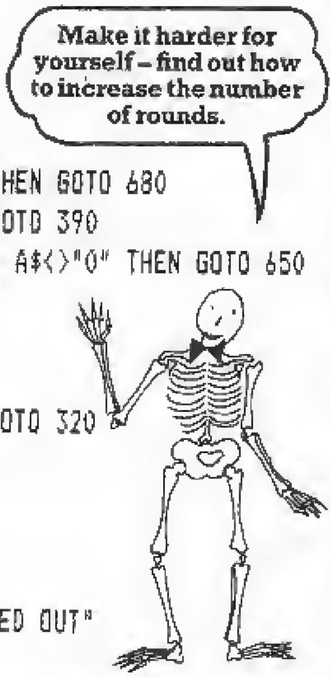
260: Goes back for next round.

270-310: Winning message if you survive 12 rounds.

330-380: Losing message.


If you keep losing, you'd better slow the program down by increasing the value of Q in line 660.
Decrease Q if the game is too easy for you.

```
   380 STOP
●▲390 CLS
■✡400 LET WX=INT(RND(1)*9+1)
■✡410 LET WY=INT(RND(1)*9+1)
   420 LET W=WX*WY
   430 LET P=P+1
   440 IF P=4 THEN GOTO 700
   450 IF P=3 THEN PRINT "YOU ARE SEEING STARS"
   460 PRINT "PANIC ON!!"
   470 PRINT
   480 PRINT "HEARTBEAT INCREASE: ";W
   490 PRINT "OXYGEN SUPPLY=";WX
   500 PRINT
   510 PRINT "AMOUNT OF ADRENALIN? ";
 ★520 LET Q=100
   530 GOSUB 580
   540 IF Z<>WY THEN GOTO 320
●▲550 CLS:IF N<11 THEN GOTO 75
   560 GOTO 270
   570 LET Q=0
 ✡580 LET Z$=""
★●▲590 LET A$=INKEY$
✡✡600 IF A$=CHR$(13) THEN GOTO 680
 ✡ 610 IF A$="P" THEN GOTO 390
   620 IF VAL(A$)=0 AND A$<>"0" THEN GOTO 650
 ✡ 630 PRINT A$;
   640 LET Z$=Z$+A$
   650 LET Q=Q+1
★●✡■660 IF Q= 500  THEN GOTO 320
   670 GOTO 590
   680 LET Z=VAL(Z$)
   690 RETURN
●▲700 CLS
   710 PRINT "YOU BLACKED OUT"
   720 STOP
```


Make it harder for yourself – find out how to increase the number of rounds.

400-410: Set oxygen supply and amount of adrenalin needed.

420: Sets heartbeat increase.

430-450: Increase panic count, print warning after 3 and end game after 4.

460-510: Print panic messages and ask for answer.

520: Sets value of Q for start of count at lines 650-660.

530: Goes to answer subroutine.

540: If answer incorrect, jumps back to losing message.

550-560: If correct, go back for next round or winning message.

570-690: ANSWER SUBROUTINE.
570: Sets count to zero.

580: Sets aside memory for Z$.

590: Checks for key press.

600: Checks if key pressed is RETURN (CHR$(13)). If so jumps down to line 680.

610: Checks if P key pressed and if so, goes to panic routine.

620: If no key pressed, jumps down to increase count Q.

630: If key other than RETURN or P pressed, it is printed.

640: Z$ is all keys pressed this round.

650-660: Time loop. If Q reaches 1000 before key is pressed, jumps up to losing message.

670: Goes back for next key press.

680: Converts numbers in string Z$ to values it can put in Z.

710: Losing message for over-use of panic.

Conversion lines

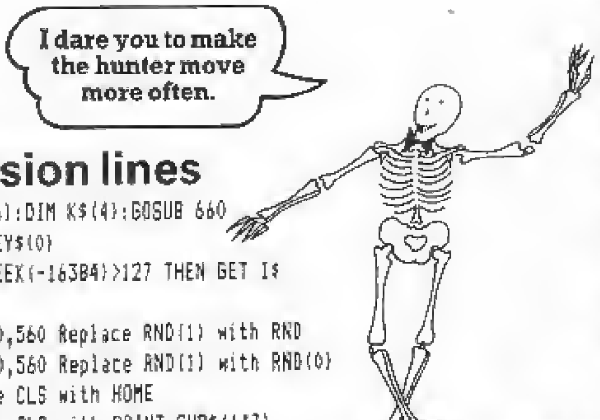
```
▲ 40,240,320,390,550,700 Change CLS to PRINT CHR$(147)
● 40,240,320,390,550,700 Change CLS toHOME
✡80,90,400,410 Change RND(1) to RND
■ 80,90,400,410 Change RND(1) to RND(0)
★520 LET Q=400
✡580 LET Z$="0"
★590 LET A$=INKEY$(1)
▲590 GET A$
●590 A$="":IF PEEK(-16384)>127 THEN GET A$
✡595 IF A$="" THEN GOTO 650
✡597 PAUSE 40
✡600 IF A$=CHR$(13) THEN PAUSE 20:GOTO 680
✡610 IF A$="P" THEN PAUSE 30:GOTO 390
✡630 PRINT A$;:PAUSE 15
★660 IF Q=1000 THEN GOTO 320
✡■●660 IF Q=250 THEN GOTO 320
```

Jaws

Mmmmmm . . . people do taste yummy! But, with The Hunter forever on your tail, it's difficult to get close enough to catch them.

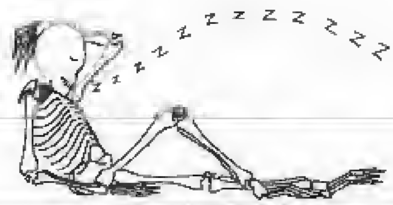
You are J (for Jaws) and you can move around by pressing keys A, Z, N and M. See how many Ps (people) you can eat before H catches you. There's a snag (of course). Each time you succeed in catching and devouring a delicious human being, you get so excited you can't remember which key does what.

```
 ✡10 DIM M$(12):DIM K$(4):GOSUB 660

   20 LET S=0:LET T=0:LET G=0

   30 LET U=1

   40 LET K$(1)="A":LET K$(2)="M"
   50 LET K$(3)="Z":LET K$(4)="N"

   60 LET PX=2:LET PY=2
   70 GOSUB 600
   80 GOSUB 490:GOSUB 440
   90 LET NX=PX:LET NY=PY
★▲●100 LET I$=INKEY$
  110 IF I$=K$(1) THEN LET NY=NY-1
  120 IF I$=K$(2) THEN LET NX=NX+1
  130 IF I$=K$(3) THEN LET NY=NY+1
  140 IF I$=K$(4) THEN LET NX=NX-1
  150 LET X=NX:LET Y=NY:GOSUB 530
  160 IF F=1 THEN GOTO 230
  170 LET X=PX:LET Y=PY:LET A$=" "
  180 GOSUB 640
  190 LET X=NX:LET Y=NY:LET A$="J"
  200 GOSUB 640
  210 LET PX=NX:LET PY=NY
  220 IF PX=TX AND PY=TY THEN GOSUB 340
  230 IF PX=GX AND PY=GY THEN GOTO 280
■✡240 IF RND(1)>U THEN GOSUB 370
  250 LET X=14:LET Y=12:LET A$=STR$(T)
  260 GOSUB 640
  270 LET G=G+1:GOTO90
  280 FOR P=1 TO 2000:NEXT P
```



Conversion lines

```
✡10 DIM M$(12,16):DIM K$(4):GOSUB 660
★100 LET I$=INKEY$(0)
●100 I$="":IF PEEK(-16384)>127 THEN GET I$
▲100 GET I$
✡240,390,400,550,560 Replace RND(1) with RND
■240,390,400,550,560 Replace RND(1) with RND(0)
●290,600 Replace CLS with HOME
▲290,600 Replace CLS with PRINT CHR$(147)
✡530 LET F=0:IF M$(Y,X)<>" " THEN LET F=1
✡640 PRINT AT Y,X;A$
●640 VTAB(Y):HTAB(X):PRINT A$
▲640 PRINT CHR$(19);:FOR LL=1 TO Y:PRINT:NEXT:PRINT TAB(X);A$
■640 PRINT@ Y*32+X,A$;
```



How the program works

10: Sets aside memory space for the grid. Goes to subroutine to read in data for grid.

20: Sets variables to zero for start.

30: Sets how often hunter moves.

40-50: Set directions in which keys move.

60: Sets starting position for Jaws.

70: Goes to subroutine which prints out grid.

80: Goes to subroutines which choose positions for hunter and person.

90: Sets Jaws' position to new position.

100-140: Calculate new position of Jaws depending on which key pressed.

150: Checks this new position is not a wall.

160: If it is a wall, don't move.

170-180: Print space at Jaws' old position.

190-200: Print J at Jaws' new position.

210: Sets Jaws' position to new position.

220: Checks if J landed on P and if so, goes to subroutine.

230: Checks if J landed on H and, if so, goes down program to end game.

240: Moves hunter.

250-260: Print score so far.

270: Increases number of moves and goes back for next one.

280: Pauses so you can see if you've been caught.

```
●▲ 290 CLS:PRINT "YOU HAVE BEEN CAUGHT!"
   300 LET S=INT(10000*(T/G)):PRINT "YOU SCORED ";S
   310 PRINT:PRINT "ANOTHER GO? (Y/N)"
   320 INPUT A$:IF A$="Y" THEN GOTO 20
   330 STOP
   340 LET T=T+1:LET U=U-0.03
   350 GOSUB 390:GOSUB 490
   360 RETURN
   370 LET X=GX:LET Y=GY:LET A$=" "
   380 GOSUB 640:GOSUB 440:RETURN
■✡390 LET K1=INT(RND(1)*4)+1
■✡400 LET K2=INT(RND(1)*4)+1
   410 LET T$=K$(K1)
   420 LET K$(K1)=K$(K2):LET K$(K2)=T$
   430 RETURN
   440 GOSUB 550
   450 IF X=TX AND Y=TY THEN GOTO 440
   460 LET GX=X:LET GY=Y
   470 LET A$="H":GOSUB 640
   480 RETURN
   490 GOSUB 550
   500 LET TX=X:LET TY=Y
   510 LET A$="P":GOSUB 640
   520 RETURN
 ✡ 530 LET F=0:IF MID$(M$(Y),X,1)<>" " THEN LET F=1
   540 RETURN
■✡550 LET X=INT(RND(1)*14)+2
■✡560 LET Y=INT(RND(1)*9)+2
   570 GOSUB 530
   580 IF F=1 THEN GOTO 550
   590 RETURN
●▲ 600 CLS:LET X=1
   610 FOR Y=1 TO 12:LET A$=M$(Y)
   620 GOSUB 640:NEXT Y
   630 RETURN
■●▲640 PRINT TAB(X,Y);A$
✡
   650 RETURN
   660 FOR I=1 TO 12:READ M$(I):NEXT I
   670 RETURN
   680 DATA "OOOOOOOOOOOOOOOO"
   690 DATA "O              O"
   700 DATA "O O OOO OO OOO O"
   710 DATA "O O O          O"
   720 DATA "O   O OOOO O O O"
   730 DATA "OOO O O    O O O"
   740 DATA "O       OO     O"
   750 DATA "O O OO O   O O O"
   760 DATA "O O OO O OOO O O"
   770 DATA "O              O"
   780 DATA "OOOOOOOOOOOOOOOO"
   790 DATA "PEOPLE EATEN:   "
```

290-330: Print message and score if Jaws is caught. Ask if player wants another game.

Skeleton speech bubble
Score depends on number of people eaten and number of moves.

340-360: Subroutine which increases score and chance of hunter moving and goes to other subroutines to change keys round and set new person position.

370-380: Subroutine which removes hunter from old position and prints him in new one.

390-430: Subroutine which changes keys around.

440-480: Subroutine for printing hunter, using another subroutine to choose a random position for him. Line 450 checks position chosen does not already have a person in it.

490-520: Subroutine for printing person, using another subroutine to choose random position.

530-540: Subroutine which checks if position X,Y coincides with a wall. If it does F is set to 1.

550-590: Subroutine which chooses a random position and checks it is not a wall.

600-630: Subroutine which prints out grid.

640-650: Subroutine which prints at position X,Y.

660-790: Subroutine which reads in data for grid.

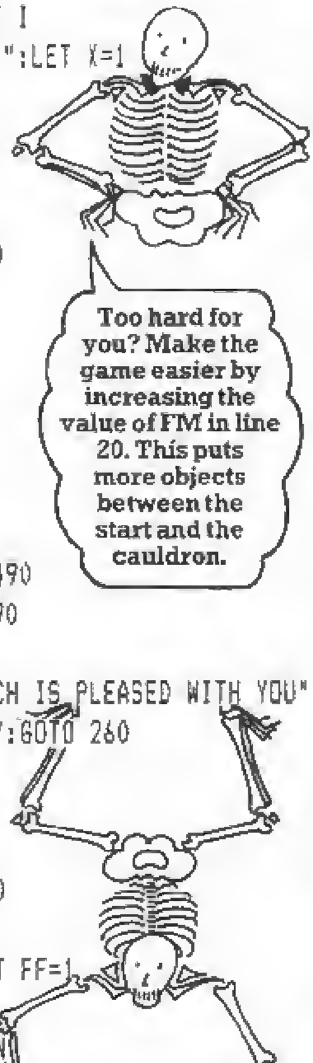
These lines make the grid. You could change it by drawing a new grid on squared paper and copying it into the program using O for walls and spaces for passages.

Flying Witches

As chief assistant to Superwitch, your job is to fly across the land on your broomstick collecting fresh ingredients for her putrid potions. You will know exactly what to collect each time because she transmits her revolting recipes directly to your computer screen.

Ingredients such as T (Toe of Newt) and B (Brain of Bat) flash past beneath you as you fly. Press any key to swoop down and pick one up (you must land right on top of it to do so). Take care, some of them could be the Broomstick Snatcher in disguise. If you land on her, you'll lose your stick. (Luckily you start the game with three spares.) You must collect everything you need before you get to the Cauldron or Superwitch will *not* be pleased with you . . .

```
 10 GOSUB 700:GOSUB 620
 20 LET NW=4:LET M=0:LET FM=50
 30 GOSUB 530:LET H=NH
 40 GOSUB 550
 50 LET EF=0:LET FF=0:LET HH=H
 60 FOR I=1 TO 4:LET G(I)=0:NEXT I
 70 FOR I=1 TO 4:LET A$=P$(I)+":":LET X=1
 80 LET Y=1+I:GOSUB 610:NEXT I
 90 GOSUB 490:GOSUB 430
100 IF M=FM THEN GOTO 250
110 IF EF=1 THEN GOTO 230
120 IF INKEY$(0)="" THEN GOTO 90
130 LET HH=H-1:GOSUB 510
140 GOSUB 490:GOSUB 430
150 LET H=HH
160 IF H>0 THEN GOTO 130
170 IF P=17 THEN GOSUB 320
180 IF FF=1 THEN GOTO 240
190 GOSUB 530
200 LET HH=H+1:GOSUB 510:GOSUB 490
210 LET H=HH:IF H=NH THEN GOTO 90
220 GOSUB 430:GOTO 200
230 LET A$="WELL DONE! SUPERWITCH IS PLEASED WITH YOU"
240 LET A$="OUT OF BROOMSTICKS!":GOTO 260
250 LET A$="TOO LATE!"
260 LET X=1:LET Y=15:GOSUB 610
270 STOP
280 LET A$="W":LET X=P:GOSUB 610
290 FOR T=1 TO 1000:NEXT T
300 LET NW=NW-1:IF NW=0 THEN LET FF=1
310 RETURN
320 IF W=1 THEN GOSUB 280:RETURN
330 LET G(R)=G(R)+1
340 LET A$=STR$(G(R)):LET X=17
350 LET Y=1+R:GOSUB 610
360 LET P=19:GOSUB 430
370 LET F=0
380 FOR I=1 TO 4
390 IF G(I)>=N(I) THEN LET F=F+1
```



Too hard for you? Make the game easier by increasing the value of FM in line 20. This puts more objects between the start and the cauldron.

Try changing the letters into graphics symbols.

Why not change the ingredients?

How the program works

20: Sets up variables. (See if you can work out what they are.)

30: H is your height above the ground. NH is your new height after a dive.

50: EF and FF are flags which are set if game is finished. HH is a variable for holding intermediate heights on the way up or down.

60: G( ) counts how many of each type of object you've picked up.

70-80: Print recipe.

90-120: Move "ground" and objects across screen until a key is pressed.

130-160: Keep ground and objects moving and move witch down to ground level.

170: If P=17 then witch has landed on object.

180: Checks flag for end of game.

200-220: Move ground and objects past until witch reaches new height.

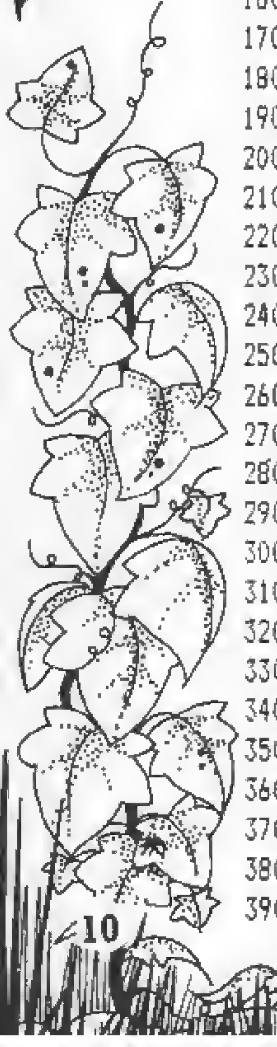
230-270: End messages.

280-310: Subroutine which shows W if object is Broomstick Snatcher in disguise. Increases NW and sets FF if all broomsticks used up.

320: Subroutine which checks if you landed on Broomstick Snatcher.

330-350: Increase number of object R collected and print new total on screen.

370-420: Check if all ingredients for this recipe have been collected.



```
 400 NEXT I
 410 IF F=4 THEN LET EF=1
 420 RETURN
 430 LET GD=GD+1:IF GD=6 THEN LET GD=1
 440 LET A$=G$(GD):LET X=1:LET Y=15:GOSUB 610
 450 LET A$=" ":LET X=P:LET Y=14:GOSUB 610
 460 LET P=P+1:IF P=20 THEN GOSUB 550
 470 LET X=P:LET A$=C$:GOSUB 610
 480 RETURN
 490 LET A$="-Y-=":LET X=16:LET Y=14-HH
 500 GOSUB 610:RETURN
 510 LET A$="    ":LET X=16:LET Y=14-H
 520 GOSUB 610:RETURN
■✡530 LET NH=INT(RND(1)*5+3)
 540 RETURN
■✡550 LET R=INT(RND(1)*4)+1
■✡560 LET W=0:IF RND(1)>0.8 THEN LET W=1
 ✡570 LET C$=LEFT$(P$(R),1):LET P=1
 580 LET M=M+1
 590 IF M=FM THEN LET C$="CAULDRON"
 600 RETURN
✡■●▲610 PRINT TAB(X,Y);A$:RETURN
●▲620 CLS:PRINT:PRINT
 630 PRINT "THE POTION MUST HAVE":PRINT
 640 FOR I=1 TO 4
■✡650 LET N=INT(RND(1)*3)+2:LET N(I)=N
 660 PRINT:PRINT;N;" ";P$(I)
 670 NEXT I:PRINT:PRINT
 680 PRINT "PRESS RETURN TO PLAY"
●▲690 INPUT X$:CLS:RETURN
 ✡700 DIM P$(4):DIM N(4):DIM G(4)
 ✡710 DIM G$(5)
 720 FOR I=1 TO 4:READ P$(I):NEXT I
 730 DATA "TOES OF NEWTS","EYES OF LIZARDS","ROOTS OF IVY","BRAINS OF BATS"
 740 LET H$=">====>====>====>====>===="
 ✡750 FOR I=1 TO 5:LET G$(6-I)=MID$(H$,I,I+20)
 760 NEXT I:LET GD=1
 770 RETURN
```

430-480: Print new section of ground and move the object along to position P. If object is past the witch (i.e. P=20, then choose a new object).

490-500: Print witch.

510-520: "Unprint" witch.

530-540: Choose new height for witch.

550: Chooses an object.

560: Random chance of object being Broomstick Snatcher.

570: Puts first letter of object in C$ and sets position P to 1.

580: Increases object counter, M.

590: If M equals maximum number of objects allowed, then C$ changed to "CAULDRON".

610: General subroutine to move cursor ready for printing.

620-690: Choose and print recipe.

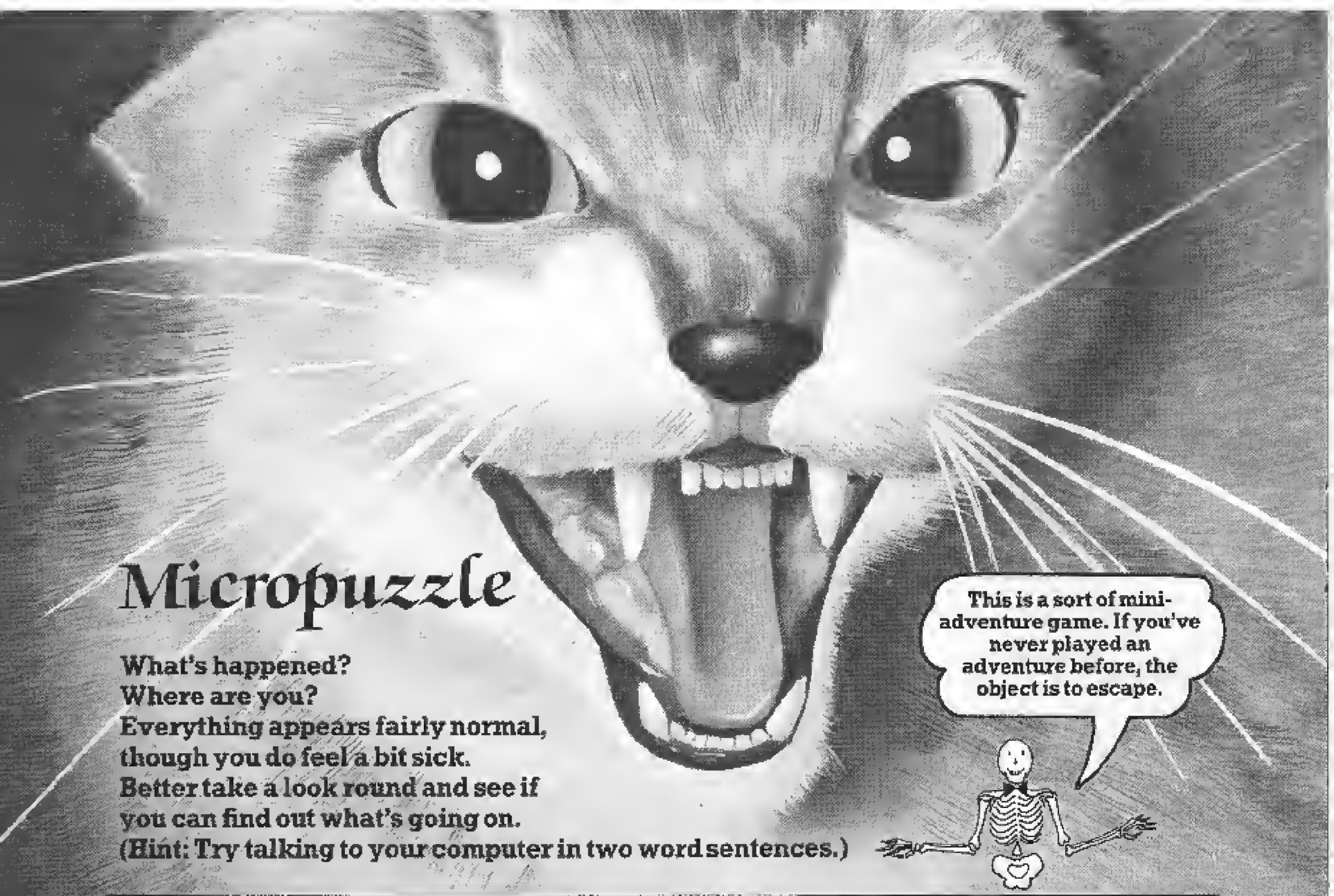
700-770: Set up data for start of program. P$ holds names of objects. H$ holds symbols for ground. Line 750 moves the > symbols along each time.

What will Superwitch do to you if you fail? Try adding comments and penalties to the game.

Conversion lines

```
■✡120 IF INKEY$="" THEN GOTO 90
 ▲120 GET I$:IF I$="" THEN GOTO 90
 ●120 I$="":IF PEEK(-16384)>127 THEN GET I$
 ●125 IF I$="" THEN GOTO 90
 ★125 *FX15,1
 ✡530,550,560,650 Replace RND(1) with RND
 ■530,550,560,650 Replace RND(1) with RND(0)
 ✡570 LET C$=P$(R,1):LET P=1
 ✡610 PRINT AT Y,X;A$:RETURN
```

```
▲610 PRINT CHR$(19):FOR LL=1 TO Y:PRINT:
     NEXT:PRINT TAB(X);A$:RETURN
●610 VTAB(Y):HTAB(X):PRINT A$:RETURN
■610 PRINT@ Y*32+X,A$;:RETURN
▲620,690 Replace CLS with PRINT CHR$(147)
●620,690 Replace CLS with HOME
✡700 DIM P$(4,15):DIM N(4):DIM G(4)
✡710 DIM G$(5,24)
✡750 FOR I=1 TO 5:LET G$(6-I)=H$(I TO I+20)
```


This is a sort of mini-adventure game. If you've never played an adventure before, the object is to escape.

Micropuzzle

**What's happened?
Where are you?
Everything appears fairly normal,
though you do feel a bit sick.
Better take a look round and see if
you can find out what's going on.
(Hint: Try talking to your computer in two word sentences.)**

How the program works

Lines 30-360: MAIN GAME SECTION

```
 10 GOSUB 1570
```

10: Goes to subroutine to read in all the data.

```
  20 RESTORE:FOR I=1 TO R:READ D$:NEXT I
●▲30 CLS:PRINT "MICRO PUZZLE"
  40 PRINT "============"
  50 PRINT "YOU ARE ";D$
```

20-50: Get a room description and print it.

```
 60 IF R=20 AND F(9)=0 THEN PRINT "YOU ARE CONFRONTED BY A LARGE CAT"
```

```
 70 FOR I=1 TO G
 80 IF L(I)=R AND F(I)=0 THEN PRINT "THERE IS A ";O$(I);" HERE."
 90 NEXT I
```

70-90: Check if there is anything there.

```
 100 PRINT:PRINT "YOU CAN GO ";
 110 FOR I=1 TO LEN(R$(R))
☆120 PRINT MID$(R$(R),I,1);",";
 130 NEXT I
```

100-130: Find which directions you can go in and print them.

```
 140 PRINT:PRINT "------------------"
```

```
 150 PRINT M$:LET M$="WHAT?"
```

150: Default message.

```
 160 IF F(16)=1 THEN PRINT "SELF DESTRUCT COUNTDOWN AT :";L
```

```
 170 PRINT "WHAT WILL YOU DO NOW":INPUT Q$
 180 LET V$="":LET W$="":LET VB=0:LET OB=0:LET LI=LEN(Q$)
☆190 FOR I=1 TO LI
☆200 IF MID$(Q$,I,1)=" " AND V$="" THEN LET V$=LEFT$(Q$,I-1)
☆210 IF MID$(Q$,I+1,1)<>" " AND V$<>"" THEN LET W$=RIGHT$(Q$,LI-I):LET I=LI
 220 NEXT I
```

170-220: Get your instructions and split them into two words.

```
✡230 IF W$="" THEN LET V$=Q$
✡240 FOR I=1 TO V
 250 IF V$=H$(I) THEN LET VB=I
 260 NEXT I
```

230-260: Check first word is a verb.

```
 270 GOSUB 440
```

270: Goes to subroutine to check second word.

```
 280 IF VB=0 THEN LET VB=V+1
 290 IF W$>"" AND OB=0 THEN LET M$="THAT IS SILLY"
 300 IF W$="" THEN LET M$="I NEED TWO WORDS"
 310 IF VB>V AND OB>0 THEN LET M$="YOU CAN'T "+Q$
 320 IF VB>V AND OB=0 THEN LET M$="YOU DO NOT MAKE SENSE"
 330 IF VB<V AND OB>0 AND OB<=G THEN LET M$="YOU DO NOT HAVE "+W$
```

280-330: Set up messages if necessary.

```
 340 LET L=L-1
```

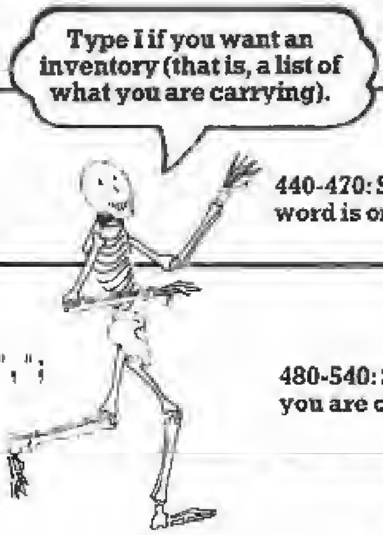
340: Counter.

```
 350 IF VB>14 THEN GOTO 380
✡360 ON VB GOSUB 480,550,550,550,550,550,550,550,760,790,790,860,920,970
 370 GOTO 390
✡380 ON VB-14 GOSUB 1080,1100,1180,1250,1320,1340,1370,1400,1460,1510,1560
```

350-380: Goes to subroutine depending on verb you typed.

```
 390 IF F(14)=0 AND L>0 THEN GOTO 20
▲400 IF L<1 THEN LET M$="YOU HAVE RUN OUT OF TIME. THE MAXIMISER SELF DESTRUCTED!"
 410 PRINT M$:PRINT "GAME OVER"
 420 IF L>0 THEN GOSUB 1510
 430 STOP
```

390-430: End of game.


Type I if you want an inventory (that is, a list of what you are carrying).

```
✡440 FOR I=1 TO W
✡450 IF W$=O$(I) THEN LET OB=I
 460 NEXT I
 470 RETURN
```

440-470: Subroutine which checks if second word is on computer's object list.

```
 480 PRINT "YOU ARE CARRYING:"
 490 FOR I=1 TO G
 500 IF L(I)=25 THEN PRINT O$(I);",";
 510 NEXT I
 520 LET M$="":PRINT
 530 GOSUB 1550
 540 RETURN
```

480-540: Subroutine which tells you what you are carrying.

```
 550 LET D=0
 560 IF OB=0 THEN LET D=VB-2
 570 IF OB>24 THEN LET D=OB-24
 580 IF R=19 AND D=6 THEN LET D=4
 590 IF R=20 AND D=6 THEN LET D=4
 600 IF R=20 AND D=5 THEN LET D=3
 610 IF F(9)=0 AND R=20 AND D=4 THEN LET M$="THE CAT WILL NOT LET YOU":RETURN
✡620 IF R=2 AND L(1)=25 AND D=3 THEN LET M$=M1$:RETURN
 630 IF R=7 AND F(3)=0 THEN LET M$="AN ANGRY MOUSE BARS YOUR WAY":RETURN
 640 LET F(13)=0:LET RL=LEN(R$(R))
 650 FOR I=1 TO RL
✡660 LET U$=MID$(R$(R),I,1)
 670 IF (U$="N" AND D=1 AND F(13)=0) THEN LET R=R-6:LET F(13)=1
 680 IF (U$="S" AND D=2 AND F(13)=0) THEN LET R=R+6:LET F(13)=1
 690 IF (U$="W" AND D=3 AND F(13)=0) THEN LET R=R-1:LET F(13)=1
 700 IF (U$="E" AND D=4 AND F(13)=0) THEN LET R=R+1:LET F(13)=1
 710 NEXT I
```

550-750: Subroutine which deals with your instructions about which direction you want to go.

If you are using a VIC 20, you will need extra memory for this game.

```
 720 LET M$="OK"
 730 IF F(13)=0 THEN LET M$="YOU CANNOT GO THAT WAY"
 740 IF D<1 THEN LET M$="GO WHERE?"
 750 RETURN
```

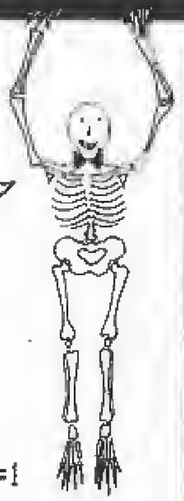
Have you played a few times and decided you'll never be able to win? You can give yourself more time if you like by putting a higher value for L in the middle of line 1910.

```
 760 LET M$="ARE YOU PRACTISING FOR THE OLYMPICS?"
 770 IF F(10)=1 AND (R=9 OR R=3) THEN LET M$="IT IS TOO FAR TO JUMP"
 780 RETURN
✡790 IF OB=8 THEN LET M$=V$+" THE "+W$+M2$:RETURN
 800 IF OB>G THEN LET M$="I CANNOT GET THE "+W$:RETURN
 810 IF L(OB)<>R THEN LET M$="IT IS NOT HERE"
 820 IF F(OB)<>0 THEN LET M$="WHAT "+W$+"?"
 830 IF L(OB)=25 THEN LET M$="YOU ALREADY HAVE IT"
✡840 IF OB>0 AND L(OB)=R AND F(OB)=0 THEN LET L(OB)=25:LET M$=M3$+W$
 850 RETURN
 860 IF L(OB)<25 THEN RETURN
 870 LET M$="NOT REALLY!":IF OB<>1 THEN RETURN
 880 PRINT "PUT KEY WHERE"
 890 INPUT W$:IF W$>"" THEN GOSUB 440
✡900 IF (OB=8 OR W$="ON "+O$(8)) THEN LET M$=M4$:LET F(15)=1:LET L(1)=R
 910 RETURN
✡920 IF F(11)=0 AND L(7)=25 THEN LET F(11)=1:LET F(3)=0:LET L(3)=0:LET M$=M5$
✡930 IF F(11)=1 AND L(7)=25 AND R=20 THEN LET F(9)=1:LET M$=M$+M6$
 940 IF R=21 AND OB=24 THEN LET F(4)=0:LET M$="DUST SETTLES"
 950 IF OB=16 THEN GOSUB 970
 960 RETURN
 970 LET M$="NOTHING OF INTEREST"
 980 IF OB=16 AND R=12 THEN LET F(2)=0:LET M$="IT IS A MICRO - VCR"
 990 IF OB=24 AND R=21 THEN LET M$="SOMETHING INSIDE"
1000 IF OB=19 AND R=24 THEN GOSUB 1400
✡1010 IF OB=23 AND R=12 THEN LET M$=M7$
1020 IF OB=7 THEN LET M$="IT CONTAINS A LARGE FLY"
1030 IF OB=9 AND R=20 AND F(9)=0 THEN LET M$="IT BITES AND SCRATCHES!"
1040 IF OB=4 THEN GOSUB 1080
1050 IF OB=1 AND L(1)=25 THEN LET M$="THE NUMBER '111' IS ENGRAVED ON IT"
1060 IF OB=6 AND L(6)=25 THEN LET M$="THERE IS A BIG RED BUTTON"
1070 RETURN
✡1080 LET M$=MG$:IF OB=4 AND L(OB)=25 THEN LET M$=MF$
1090 RETURN
1100 IF OB<>5 THEN LET M$="CANNOT TIE "+W$:RETURN
1110 IF L(5)<25 THEN RETURN
1120 LET W$="":PRINT "TIE THE THREAD TO WHAT"
1130 INPUT W$
1140 IF W$>"" THEN GOSUB 440
1150 LET M$="CANNOT TIE IT TO "+W$
✡1160 IF OB=13 AND R=9 THEN LET F(5)=1:LET L(5)=0:LET M$=M8$
1170 RETURN
1180 IF OB=5 AND F(5)=0 THEN LET M$="IT IS NOT TIED TO ANYTHING!"
1190 IF OB=5 AND R=9 AND F(5)=1 THEN LET R=8:LET F(10)=0:LET M$="OK":RETURN
1200 IF OB=5 AND R=8 AND F(5)=1 THEN LET R=9:LET F(10)=1:LET M$="OK"
1210 IF OB=13 AND R=9 THEN LET M$="IT IS TOO SMOOTH TO CLIMB"
1220 IF OB=W AND R=19 THEN LET OB=30:GOSUB 550
```

760-1500: These are subroutines for the verbs used in the program. Look for the lines which say RETURN to see where one ends and another starts.

Unless you are using a Spectrum, you can leave out all the LETs if you like.

```
 1230 IF OB=W AND R=20 THEN LET OB=29:GOSUB 550
 1240 RETURN
 1250 IF OB<>6 OR L(6)<>25 THEN RETURN
 1260 LET W$="":PRINT "POINT IT AT WHAT"
 1270 INPUT W$
 1280 IF W$>"" THEN GOSUB 440
 1290 IF OB=22 THEN LET F(6)=1
 1300 LET M$="VERY WELL"
 1310 RETURN
 1320 IF OB=3 AND L(OB)=25 THEN LET L(OB)=0:LET M$="MUNCH CHOMP"
 1330 RETURN
▲✡1340 IF (OB=19 OR OB=17) AND R=24 AND L(1)<25 THEN LET M$=M9$:LET L=L-12:LET F(16)=1
 ✡1350 IF (OB=19 OR OB=17) AND R=24 AND L(1)=25 THEN LET F(12)=1:LET M$=MH$
 1360 RETURN
▲✡1370 IF OB=3 AND R=7 AND L(OB)=25 THEN LET F(3)=1:LET L(OB)=0:LET B=B+1:LET M$=MA$
 1380 IF L(OB)=25 THEN LET L(OB)=R:LET M$="DONE"
 1390 RETURN
 1400 IF R<>24 THEN LET M$="NOTHING HERE TO TYPE ON!":RETURN
 1410 IF F(12)=0 THEN LET M$="THIS TERMINAL IS NON-ACTIVATED":RETURN
 1420 LET M$="THE TERMINAL ECHOES :"+W$
 1430 IF OB=18 THEN LET M$="CODEWORD ACCEPTED":LET F(17)=1
 1440 IF OB=20 THEN LET M$="TERMINAL ID"
 1450 RETURN
 ✡1460 IF R=14 AND OB=11 AND L(8)=22 THEN LET L(8)=14:LET M$=MB$+MC$:RETURN
 ✡1470 IF R=14 AND OB=11 AND L(8)=14 THEN LET L(8)=22:LET M$=MB$+MD$
 1480 IF F(15)=1 THEN LET L(1)=L(8)
▲✡1490 IF OB=21 AND F(6)=1 AND L(6)=25 AND R=18 AND F(17)=1 THEN LET B=B+10:LET M$=ME$:LET F(14)=1
 1500 RETURN
```



This program is not explained as fully as some of the others so as not to give you too many clues about the game. Try working out what all the lines do after you have played a few times.

```
 1510 LET S=0:FOR I=1 TO G
 1520 IF L(I)=25 THEN LET S=S+I
 1530 NEXT I
 1540 LET S=S+B:PRINT "YOUR SCORE = ";S
 1550 IF F(14)=0 THEN PRINT "PRESS RETURN TO CONTINUE":INPUT Q$
 1560 RETURN
```

1510-1560: Scoring subroutine.

```
 1570 LET V=24:LET W=31:LET G=8
 ✡1580 DIM R$(24):DIM O$(W):DIM H$(V)
 1590 DIM L(G):DIM F(17)
```

1570-1590: Set up variables and dimension arrays.

```
 1600 DATA "INSIDE THE MOUSEHOLE - IT IS VERY DARK IN HERE"
 1610 DATA "AT A MOUSEHOLE IN A CORNER OF THE ROOM","ON THE EDGE OF A HIGH TABLE"
 1620 DATA "AT THE BACK OF A HALLWAY","IN A STORAGE ROOM","IN THE KITCHEN"
 1630 DATA "FURTHER DOWN A DARK SMELLY TUNNEL","BY A RAILWAY SIDING"
 1640 DATA "AT THE BASE OF A TALL PLASTIC TREE ON THE EDGE OF A HIGH TABLE"
 1650 DATA "OUTSIDE THE OPEN DOOR OF AN ODDLY PROPORTIONED HOUSE"
 1660 DATA "IN A YELLOW FRONT ROOM","BY A TV SET AND A RECORDER"
 1670 DATA "AT THE END OF A DARK TUNNEL"
 1680 DATA "BY A LARGE SWITCH CONNECTED TO THE RAILWAY TRACKS"
 1690 DATA "ALONGSIDE THE WINDING TRACK"
▲1700 DATA "AT THE END OF THE LINE-THE TRACK DISAPPEARS THROUGH A HOLE IN THE WALL"
 1710 DATA "BELOW A WHOLE WALL OF OVERSIZED VIDEO SCREENS"
 1720 DATA "STANDING ON THE MAXIMISER PAD"
 1730 DATA "ON A SHELF OF DISTURBING APPARATUS - THERE IS A STOOL NEARBY"
 1740 DATA "ON A SHORT STEP STOOL"
```

1600-1870: Data lines

```
 1750 DATA "ON THE FLOOR OF AN OVERTURNED BOX OF BROKEN ELECTRONIC PARTS"
 1760 DATA "AT A HOLE IN THE WALL FROM WHICH A RAILWAY LINE EMERGES"
 1770 DATA "AT THE BASE OF A SWIVEL CHIAR"
 1780 DATA "STANDING ON A COMPUTER TERMINAL WITH A SECRURITY LOCK"
 1790 DATA 16,12,3,21,5,17,19,14
 1800 DATA "I","GO","N","S","W","E","U","D","JUMP","GET","TAKE","PUT","OPEN"
 1810 DATA "EXAMINE","READ","TIE","CLIMB","POINT","EAT","UNLOCK","LEAVE","TYPE"
 1820 DATA "PRESS","SCORE","SE","SW","SE","SWE","WE","SW","NS","NS","NE","NW"
 1830 DATA "EW","NW","NS","NE","EW","W","S","S","NED","EWUD","EW","EW","NEW"
 1840 DATA "NW","KEY","CASSETTE","CHEESE","PAPER","THREAD","REMOTE-CONTROL"
▲1850 DATA "BOTTLE","TRAIN","CAT","DOOR","SWITCH","TUNNEL","TREE","HOLE","MOUSE"
 1860 DATA "VIDEO","COMPUTER","GROCER","TERMINAL","111","BUTTON","MAXIMISER"
 1870 DATA "TV","BOX","NORTH","SOUTH","WEST","EAST"."UP","DOWN","STOOL"
```

```
 1880 FOR I=1 TO 24:READ D$:NEXT I:FOR I=1 TO G:READ L(I):NEXT I
 1890 FOR I=1 TO V:READ H$(I):NEXT I:FOR I=1 TO 24:READ R$(I):NEXT I
 1900 FOR I=1 TO W:READ O$(I):NEXT I:LET F(10)=1:LET F(4)=1:LET F(2)=1
```

1880-1900: Read in data.

```
✡1910 LET R=11:LET B=8:LET L=100:LET M$="YOU AWAKEN..":LET MH$="TERMINAL ACTIVE"
✡1920 LET M1$="YOU CANNOT TAKE THE KEY THROUGH":LET M2$=" EH? VERY FUNNY!"
✡1930 LET M3$="YOU HAVE THE ":LET M4$="WELL DONE!":LET M6$="NOTHING OF INTEREST"
✡1940 LET M5$="A LOUDLY BUZZING FLY FLIES OUT":LET MB$="THE TRAIN CHUGS "
✡1950 LET M7$="IT IS JUST A BOX WITH PHOTO STUCK ON":LET M9$="*! TAMPER *!"
✡1960 LET M8$="IT IS SECURELY TIED.":LET M6$=" AND THE CAT CHASES AFTER IT!"
✡1970 LET MA$="THE MOUSE RUNS OFF WITH IT.":LET MF$="TERMINAL PASSWORD="+O$(18)
✡1980 LET MC$="INTO SIGHT AND STOPS HERE":LET MD$="AWAY AND INTO A TUNNEL"
✡1990 LET ME$="THE MAXIMISER BEAM WORKS. YOU ARE RETURNED TO NORMAL SIZE"
 2000 RETURN
```

1910-2000: Set up messages.

Conversion lines

```
▲ 30 Replace CLS with PRINT CHR$(147)
● 30 Replace CLS with HOME
```

IMPORTANT NOTE for Commodore users. If you are using a C64 or a VIC 20, you must leave out the LETs in lines 400, 1340, 1370 and 1490. If you are using a C64, you must leave out all the spaces in lines 1490, 1700 and 1850.

. . . FOR SPECTRUM USERS ✡

```
 120 PRINT R$(R,I);",";
 190 FOR I=1 TO LI-1
 200 IF Q$(I)=" " AND V$="" THEN LET V$=Q$( TO I-1)
 210 IF Q$(I+1)<>" " AND V$<>"" THEN LET W$=Q$(I+1 TO ):LET I=LI-1
 235 IF LEN(V$)<7 THEN LET V$=V$+" ":GOTO 235
 360 GOSUB 480*(VB=1)+550*(VB>1 AND VB<9)+760*(VB=9)+790*(VB=10 OR VB=11)
     +860*(VB=12)+920*(VB=13)+970*(VB=14)
 380 GOSUB 1080*(VB=15)+1100*(VB=16)+1180*(VB=17)+1250*(VB=18)+1320*(VB=19)+1340*(VB=20)
     +1370*(VB=21)+1400*(VB=22)+1460*(VB=23)+1510*(VB =24)+1560*(VB=25)
 440 IF LEN(W$)<14 THEN LET W$=W$+" ":GOTO 440
 445 FOR I=1 TO W
 660 LET U$=R$(R,I)
 900 IF (OB=8 OR W$="ON "+O$(B)+"         " THEN LET M$="WELL DONE!":LET F(15)=1:LET L(1)=R
1580 DIM R$(24,4):DIM O$(W,14):DIM H$(V,7)
1910 LET R=11:LET B=8:LET L=100:LET M$="YOU AWAKEN.."
Lines 620,790,840,920,930,1010,1080,1160,1340,1350,1370,1460,1470,1490
Replace M1$ to MH$ in these lines with the text in lines 1910 to 1990
eg. 620 IF R=2 AND L(1)=25 AND D=3 THEN LET M$="YOU CANNOT TAKE THE KEY THROUGH":RETURN
Leave out lines 1920 to 1990
```

Answers to puzzles

Here are the answers to some of the puzzles in this book.

Skulls of The Pyramid

To add more skulls, change the 5 in line 260 to a higher number.

The limits of the number to be scored are set in line 20. As the program stands the number chosen is between 10 and 39. If you change the 30 to 40, you will get a number between 10 and 49. If you also change the 10 to 20, you will get a number between 20 and 49.

Monster Wrestling

To increase the number of rounds, change the number 11 in lines 260 and 550 to a higher number. (It must be the same in both.)

If you increase the number of rounds, you may want more panics. To get these, change the number 4 in line 440 to a higher number.

Jaws

To make the hunter move more often, decrease the value of U in line 30, for example to 0.8.

Flying Witches

If you fail the program goes either to line 240 or 250 for a message. Change these to whatever you like.

Usborne Computer Books

"Highly recommended to anyone of any age." Computing Today

"Without question the best general introduction to computing I have ever seen." Personal Computer World

". . . perhaps the best introduction around . . . outstanding . . ." Educational Computing

"These books are outstanding . . . they make all other young people's computer books look meretricious." Times Educational Supplement

Guide to Computers
Understanding the Micro
Computer and Video Games
Computer Jargon
Computer Graphics
Inside the Chip
Computer Programming
Practise Your BASIC
Better BASIC

Machine Code for Beginners
Practical Things to do with a Microcomputer
Computer Spacegames
Computer Battlegames
Write Your Own Adventure Programs
Creepy Computer Games

New Titles

Programming Tricks & Skills Professional tips and tricks for better programming.

Experiments With Your Computer An exciting introduction to scientific investigation with a microcomputer.

Expanding Your Micro A detailed guide to add-ons and interfaces.

Write Your Own Fantasy Games A step-by-step guide to writing fantasy games, with program listing.

Weird Computer Games, Computer Spy Games Short listings to run on most main home computers.

Mystery of Silver Mountain, Island of Secrets Unique adventure game books containing full program listings.

ISBN 0-86020-827-3

9 780860 208273

Printed in Belgium

99p